An Accurate Model for Fair Rate Calculation in Resilient Packet Rings

Arash Shokrani, Ioannis Lambadaris, and Jérôme Talim
Broadband Networks Laboratory
Department of Systems and Computer Engineering, Carleton University, Ottawa, Canada
{shokrani, ioannis, jtalim}@sce.carleton.ca

Abstract

Resilient Packet Ring (RPR), which is being standardized as IEEE 802.17, is a new Medium Access Control (MAC) protocol for high-speed metro-area ring networks. RPR supports spatial reuse and therefore requires to ensure fairness among different nodes competing for the ring bandwidth. In order to achieve fairness among nodes, a fairness algorithm is employed at each RPR node. In case of congestion, the fairness algorithm calculates and advertises a fair rate to all upstream nodes contributing to the congestion point. Consequently, the congested node will be able to insert its local traffic into the ring. In this paper, we develop an accurate model for fair rate calculation in the standard RPR fairness algorithm. We first ignore the link propagation delay and model the system using a non-linear discrete-time low-pass filter. We then consider link propagation delay and develop a more realistic model. We verify our model by simulation results and analyze the effect of system parameters on the calculated fair rate. This model can be used to evaluate performance of the RPR standard fairness algorithm in terms of stability and convergence time.

Keywords—*Resilient Packet Ring (RPR), fairness algorithm, parking lot scenario, discrete-time nonlinear low-pass filter.*

1 Introduction

The Resilient Packet Ring (RPR) IEEE 802.17 standard is a new Medium Access Control (MAC) protocol for metro-ring networks [1]. RPR supports *spatial reuse*, that is, multiple concurrent transmissions over different parts of the ring [2]. However, it may cause congestion and unfairness among different nodes in accessing the ring bandwidth. Hence a bandwidth allocation algorithm is required in order to provide a fair ring access to all nodes.

The ring access scheme in RPR is based on Buffer Insertion Ring (BIR) method, in which the transit traffic at each node can block the local traffic of that node in accessing onto the ring [3]. Therefore, all nodes should be forced to adjust the insertion rate of their local traffic (add-rate) according to their fair shares.

The objective of the fairness algorithm in RPR is to fairly distribute the available bandwidth on any link between the local traffic of all competing nodes on that link. The RPR bandwidth management is based on the concept of explicit rate feedback. When a node is congested, it calculates a first approximation to the true fair rate, which is advertised to the upstream nodes through a control message. As the upstream nodes receive the control message, they adjust their add-rates according to the received fair rate. Notice that as the calculated fair rate is an approximation to the true fair rate, the congestion may not be cleared at the congested node. In this case, the congested node calculates another approximation of the fair rate and advertises it to the upstream nodes. This process may repeat until the calculated fair rate converges to the true fair rate. The time it takes for the fairness algorithm to converge is commonly referred to as the *convergence time*, which is an important criterion in evaluating performance and efficiency of an RPR fairness algorithm. In this paper, we present an *accurate* analytical model for fair rate calculation in the standard RPR fairness algorithm. This analysis is done for the parking lot scenario depicted in Figure 1. In this scenario all nodes 1, 2, $\cdots$, and n send traffic to node 0 (Hub), sharing the link between node 1 and node 0. This scenario is one of the traditional and challenging benchmark scenarios in RPR, and is studied in standardization process of the RPR, as well as in [4], to compare performance of different RPR fairness algorithms.

The rest of this paper is organized as follows. In Section 2 the RPR standard fairness algorithm is described. In Section 3, we present our analytical model of fair rate calculation. Section 4 deals with the numerical results. Finally, some conclusions are given in Section 5.

This work was supported by a grant from Communications and Information Technology Ontario (CITO)

2 RPR Standard Fairness Algorithm

At each RPR node, the arriving traffic from the ring is dropped if destined to that node. Otherwise, it is forwarded to a transit buffer. When the transit buffer of a node is implemented with a single queue, the ring traffic in the transit buffer has non-preemptive priority over local traffic of that node. As a result, the ring traffic transiting a node can block the local traffic of that node. When a node is not given access onto the ring, it is starved and congested.

Congestion detection and fair rate calculation at each RPR node are performed at the end of every fixed time interval called *control interval*. Each RPR node measures the service rate of its local traffic (called *add-rate*) and that of the transit traffic (called *forward-rate*) in each control interval. Note that the bandwidth for high priority traffic is reserved in RPR and we do not consider it in this paper. Let us define the following notations:

- C is the available bandwidth on a link.
- Control interval $k > 0$ with length of T seconds, is the time interval $\big((k-1)\cdot T,\, k\cdot T\big]$.
- α is a low-pass filtering coefficient ($0 < \alpha < 1$).
- $r_i^a(k)$ is the add-rate of node i in control interval k and $\tilde{r}_i^a(k)$ is its low-pass filtered version defined as follows:

$$\tilde{r}_i^a(k) = (1-\alpha)\cdot\tilde{r}_i^a(k-1) + \alpha\cdot r_i^a(k). \qquad (1)$$

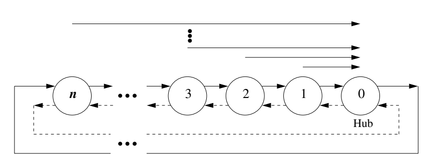

Figure 1. Parking lot scenario

• $r_i^f(k)$ is the forward-rate of node i in control interval k and $\tilde{r}_i^f(k)$ is its low-pass filtered version obtained similar to (1).
• $u_i(k) = r_i^a(k) + r_i^f(k)$ is the usage rate of the output link of node i in control interval k, and $\tilde{u}_i(k)$ is its low-pass filtered version. Intuitively, we have

$$\tilde{u}_i(k) = \tilde{r}_i^a(k) + \tilde{r}_i^f(k) = (1-\alpha)\cdot\tilde{u}_i(k-1) + \alpha\cdot u_i(k). \quad (2)$$

For simplicity, assume that all rates are normalized to C. Therefore, $u_i(k) \leq 1$ and $r_i^a(k) \leq 1 - r_i^f(k)$.

1) Congestion Detection: Assume that the transit buffer of node i is configured in single-queue mode. At the end of control interval k, node i detects congestion if $\tilde{u}_i(k)$ exceeds a certain threshold τ (e.g., $0.95{\cdot}C$); or a packet at the head of local buffers of that node experiences a long delay to get onto the ring.

2) Fair Rate Calculation: The RPR standard fairness algorithm operates in two modes: *Aggressive Mode* (RPR-AM) and *Conservative Mode* (RPR-CM) [5]. We assume that all nodes are operating in RPR-AM, as this is the default mode of operation in RPR. In RPR-AM, when node i detects congestion in control interval k, it advertises $\tilde{r}_i^a(k)$ as the fair rate to its upstream nodes. If node i is not congested, the fair rate is set to 1.

3) Control Message Transmission: We assume that there is a single control message circulating along the ring within each control interval. The control message contains separate fields for the calculated fair rate of all nodes. At the end of each control interval, one of the nodes initiates transmission of the control message. It inserts its advertised fair rate in a designated field and sends the control message to its upstream node. Upon receiving a control message, node i adjusts the rate of its local traffic so that its add-rate for destinations, which are in the downstream of node j, does not exceeds the advertised fair rate of node j. It also updates the corresponding field in the control message with its most recently calculated fair rate and forwards the message to its upstream node.

In the next section we present an analytical model for fair rate calculation in RPR-AM.

3 Problem Description

We consider parking lot scenario depicted in Figure 1, which is a benchmark scenario for evaluating performance of an RPR fairness algorithm. We assume that nodes 1, 2, $\cdots$, and n in Figure 1 have infinite amount of low priority traffic in their local buffers destined to node 0. Also, we do not consider high priority traffic as the fairness algorithm only controls the rate of low priority traffic throughout the ring. All nodes compete for the bandwidth on the link between node 1 and node 0. As a result, this link becomes the most congested link. We assume that all nodes have equal weights. Therefore, the fair share of each node is $\frac{1}{n}$ of the link bandwidth. We also assume that the fairness algorithm is RPR-AM and transit queue at each node is implemented in single queue mode. In Figure 1, the solid-line ringlet is the traffic path to node 0. The control message is transmitted backward on the dashed-line ringlet.

In the following, we first assume that the propagation delay of the links is zero and investigate the convergence of the fairness algorithm. Then, we incorporate a deterministic link propagation delay into the model and analyze the problem in a more realistic case.

3.1 Analysis of the Model Without Link Propagation Delay

In this section we ignore the link propagation delay and model the fair rate calculation in RPR-AM. Let us define $\ell_i(k)$ as the rate limiter value of the local traffic of node i at the beginning of control interval k. This is the maximum rate at which node i is allowed to add traffic to the ring destined to node 0. The add-rate of node i in control interval k is also bounded by $1 - r_i^f(k)$ and hence we have

$$r_i^a(k) = \min\left\{1 - r_i^f(k), \ell_i(k)\right\}. \quad (3)$$

1) Congestion Detection Time: Starting at time 0, each node has traffic at link rate to transmit to the hub. The initial rate limiter value of all nodes is equal to the link rate: $\ell_i(1) = 1$, $i = 1,2,\cdots,n$. Note that at each node the transit traffic has priority over local traffic of that node. As a result, only node n, which is the most upstream node, can transmit at the link rate, i.e., $r_n^a(k) = 1$; other nodes will not be able to add their local traffic to the ring before congestion detection. Hence, $r_i^a(k) = 0$, $i = 1,\cdots,n-1$. Note that as node n sends traffic at like rate to node 0, before congestion detection we have $u_i(k) = 1$, $i = 1,\cdots,n$.

We assume that congestion is triggered at node i only when $\tilde{u}_i(k)$ exceeds τ. As congestion is detected through a low-pass filtering approach, it takes several control intervals for all nodes to detect congestion. For node 1, in particular, we have $\tilde{u}_1(0) = 0$ and $u_1(k) = 1$, for all k prior to congestion detection. Therefore, one can derive $\tilde{u}_1(k)$, using (2), as follows:

$$\tilde{u}_1(k) = 1 - (1-\alpha)^k. \quad (4)$$

Node 1 detects congestion at the end of control interval $\hat{k}$, where $\hat{k} = \min\left\{k : \tilde{u}_1(\hat{k}) \geq \tau\right\}$. Using (4) we obtain

$$\hat{k} = \left\lceil \frac{\ln(1-\tau)}{\ln(1-\alpha)} \right\rceil, \quad (5)$$

where $\lceil x \rceil$ is the smallest integer number which is greater than or equal to x. Note that nodes 2, 3, $\cdots$, and n, detect the congestion at the same time as node 1.

2) Advertised Fair Rate: Assume that node 1 initiates the control message transmission at the end of each control interval. As the size of control message in RPR is relatively short, we can ignore the transmission and the processing time of the control message. Moreover, we have assumed that link propagation delay is zero in this ideal model. As a result, the time it takes for a control message to reach the furthest upstream node in the ring (i.e., node n), is zero. This means that all upstream nodes of node 1 receive the control message right after it is transmitted. Prior to the control interval $\hat{k}$, only node n can add traffic to the ring. Therefore, we have

$$\tilde{r}_i^a(\hat{k}) = 0, \quad i = 1,2,\cdots,n-1. \quad (6)$$

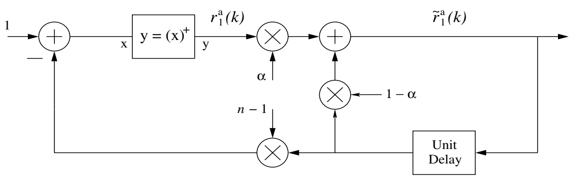

Figure 2. Low-pass filter model of fair rate calculation.

When congestion is detected, node 1 advertises its calculated fair rate, i.e., $\tilde{r}_1^a(\hat{k}) = 0$, to its upstream nodes. When node i, $i > 1$, receives the control message, it sets its rate limiter value to the received fair rate, i.e. $\ell_i(\hat{k}+1) = \tilde{r}_1^a(\hat{k}) = 0$. Note that the rate limiter value of a node only changes when that node receives a congestion message from one of its downstream nodes. Therefore, $\forall k$, $\ell_1(k) = 1$, as there is no congested node on downstream of node 1. Consequently, from (3) we have

$$\forall k, r_1^a(k) = \min\left\{1 - r_1^f(k), \ell_1(k) = 1\right\} = 1 - r_1^f(k). \quad (7)$$

As $\forall i$, $u_i(k) = r_i^a(k) + r_i^f(k)$, it follows that $\forall k$, $u_1(k) = 1$. Therefore, $\tilde{u}_1(k)$ can always be obtained from (4). Note that $\tilde{u}_1(k)$ in (4) is an increasing function of k and consequently $\tilde{u}_1(k)$ will be greater than τ for all $k \geq \hat{k}$. This means that node 1 is congested for all $k \geq \hat{k}$. As a results, node 1 calculates and advertises a fair rate to the upstream nodes in every control interval after congestion detection. We need to know the add-rate od node 1 in order to calculate the new fair rate. In general, in control interval $k > \hat{k}$, $\ell_i(k) = \tilde{r}_1^a(k-1)$, $i > 1$, $r_1^f(k) = \min\left\{1, (n-1)\cdot\tilde{r}_1^a(k-1)\right\}$, and the add-rate of node 1 is obtained as

$$r_1^a(k) = \left(1 - (n-1)\cdot\tilde{r}_1^a(k-1)\right)^+, k > \hat{k}, \quad (8)$$

where $(x)^+ = \max\left\{0, x\right\}$. The fair rate of node 1 at the end of control interval $k > \hat{k}$ can be calculated as follows:

$$\tilde{r}_1^a(k) = (1-\alpha)\cdot\tilde{r}_1^a(k-1) + \alpha\cdot\left(1-(n-1)\cdot\tilde{r}_1^a(k-1)\right)^+. \quad (9)$$

It can be seen that $\tilde{r}_1^a(k)$ depends on α, n, and the previous calculated fair rate, $\tilde{r}_1^a(k-1)$. Therefore, we can model the fair rate calculation process at node 1 using the first-order non-linear discrete-time filter depicted in Figure 2, where the unit delay is equal to one control interval. The advertised fair rate from node 1 is the response of the filter to a unit step function applied at $k = \hat{k}+1$, with initial condition $\tilde{r}_1^a(\hat{k}) = 0$.

3) Condition for Finite Convergence Time of the Fair Rate: Consider the filter in Figure 2 and assume that the non-linear block is bypassed (i.e., $y = (x)^+$ replaced by $y = x$). In this case, the system is a first-order *linear* discrete-time filter with transfer function

$$H(z) = \frac{\alpha}{1 + (\alpha\cdot n - 1)\cdot z^{-1}}\,. \quad (10)$$

This linear system is stable when its pole is inside the unit circle, which yields

$$\alpha < \frac{2}{n}. \quad (11)$$

We have analytically proved in [6] that the stability condition of the non-linear filter is given by (11) as well. Moreover, we have shown that the response of the non-linear stable system to the unit step function converges to $\frac{1}{n}$, i.e., the true fair rate.

Note that in this model, node 1 is assumed to be the most congested node and all other nodes adjust their local rates based on the advertised fair rate of node 1. To maintain this condition, the congestion threshold, τ, should be greater than $1 - \frac{1}{n}$ and close to 1. In this case, the other nodes are not congested for $k > \hat{k}$, and always set their rates to the advertised fair rate of node 1.

3.2 Analysis of the Model With Link Propagation Delay

We now consider the effect of link propagation delay in calculating the fair rate. Up to the congestion detection, this system works like the ideal model and congestion detection time can still be calculated from (5). We define d as the one-hop *round-trip* propagation delay. In this case the propagation delay of each link is equal to $\frac{d}{2}$. Figure 3 illustrates the effect of the link propagation delay on the add-rate of node 1 in a scenario with $n = 3$. It presents a typical control interval k, at which node 1 sends a control message to its upstream nodes. In order to simplify the problem, we assume that the time it takes for the congested node 1 to see the traffic change of node n, which is the furthest node contributing to congestion, is less than the length of control interval. That is,

$$(n-1)\cdot d < T. \quad (12)$$

Therefore, after advertising a new fair rate at the beginning of each control interval, congested node 1 can see the effect of the traffic change of all nodes contributing to congestion in the same control interval. As node 1 gradually sees the effect of traffic changes of its upstream nodes, its add-rate changes during a control interval. In this case, the average add-rate of node 1 during a control interval should be measured in order to calculate the fair rate. In [6] we show that the average add-rate of node 1 in control interval $k > \hat{k}$ is calculated as

$$r_1^a(k) = \left(1 - (n-1)\cdot\frac{d}{T}\right)\cdot r_1^a(k,n) + \frac{d}{T}\cdot\sum_{j=1}^{n-1} r_1^a(k,j), \quad (13)$$

where $r_1^a(k,j) = \left(1-(n-j)\cdot\tilde{r}_1^a(k-2)-(j-1)\cdot\tilde{r}_1^a(k-1)\right)^+$, $1 \leq j \leq n$.

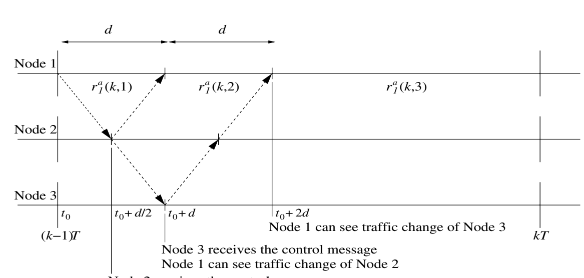

Figure 3. Effect of propagation delay on advertising the fair rate.

The calculated fair rate of node 1 in control interval k, $\tilde{r}_1^a(k)$, is obtained by substituting (13) in (1). One can see that $\tilde{r}_1^a(k)$, $k > \hat{k}$, depends on α, n, $\tilde{r}_1^a(k-2)$, and $\tilde{r}_1^a(k-1)$. As a result, using a method similar to the one applied to the ideal case ($d = 0$), we can model the fair rate calculation at node 1 by a second-order nonlinear discrete-time filter.

We have investigated and proved several properties of this model in [6]. We observed that the stability domain of the second-order nonlinear filter is similar to the corresponding linear filter, which is obtained by bypassing all non-linear blocks in (13). Also, we proved that the linear system with propagation delay is less sensitive to the value of α compared to the linear system without propagation delay. Finally, we showed that the maximum stability domain for the linear system with delay is achieved when $(n-1) \cdot d = 0.5 \cdot T$.

4 Numerical Results and Analysis

In this section we compare the results from the analysis with the ones from simulations to evaluate the accuracy of the model. We also study the effect of the low-pass filtering coefficient and the propagation delay on the calculated fair rate. The simulation model is implemented in OPNET. We consider 100Mbps links, 512 bits packet size, and $T = 1$msec.

1) Effect of α on Fair Rate Oscillations: The oscillations of the fair rate result from an overestimation or underestimation of the rates. As the oscillations contribute to the convergence time, they are part of the performance of the RPR fairness algorithm. Figure 4 presents both analytical and simulation results of the calculated fair rate in a scenario with $n = 8$, $d = 0.1T$, $\tau = 0.95$, for $\alpha = 0.05$, 0.15, and 0.25. One can first note that analytical and simulation results are very close. In fact, the maximum relative error in all cases is less than 2%. Our analytical model is a very accurate approximation of the real system, and therefore may be used to evaluate the performance of the RPR fairness algorithm. In Figure 4, congestion is detected at control intervals 59, 19, and 11 for $\alpha = 0.05$, 0.15, and 0.25, respectively. One can see that as α increases, congestion detection time decreases. However, the calculated fair rate converges to the true fair rate with a larger number of oscillations. The reason is that when α is large, the latest measured data has more weight in the calculation of the next fair rate, according to (1), and the true fair rate may not be estimated accurately. Thus, the fair rate may be overestimated, then underestimated, and so on.

2) Effect of Propagation Delay on Fair Rate Oscillations: Figure 5 illustrates the effect of propagation delay on the fair rate calculation for $n = 4$ and $\alpha = 0.5$. Both analytical and simulation results are shown. In this case congestion is detected at the end of the 5^{th} control interval. As delay increases, the congested node sees the effect of traffic changes due to the last advertised rate in a smaller portion of the control interval, which changes the value of the calculated rate. For example for $d = 0.2T$, the first overshoot of the fair rate occurs with one control interval delay comparing with the other two cases due to the propagation delay. The maximum relative error of the simulation results in all cases is less than 1.5%.

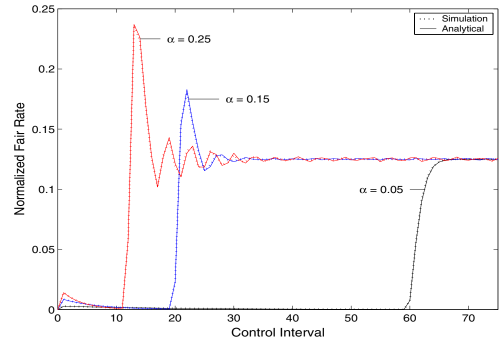

Figure 4. Normalized calculated fair rate for $n = 8$, $\tau = 0.95$, and $d = 0.1T$.

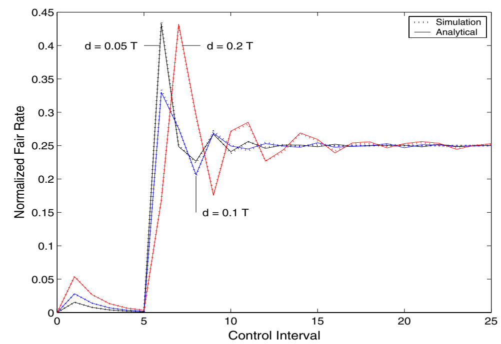

Figure 5. Normalized calculated fair rate for $n = 4$, $\tau = 0.95$, and $\alpha = 0.5$.

5 Conclusion

We develop an analytical model for fair rate calculation in standard RPR fairness algorithm in the parking lot scenario. Our approach is to model the fair rate calculation process as a nonlinear discrete-time low-pass filter. First, we ignore the link propagation delay and derive the fair rate equation. We then consider the propagation delay and develop a more realistic model. Simulation results confirm the model accuracy. Using this model, we study the effect of the system parameters on the calculated fair rate. This model can be used to evaluate performance of the RPR fairness algorithm in terms of stability and convergence time.

References

[1] F. Davik, M. Yilmaz, S. Gjessing, and N. Uzun, "IEEE 802.17 Resilient Packet Ring Tutorial," *IEEE Communication Magazine*, vol. 42, no. 3, pp. 112-118, March 2004.

[2] I. Cidon, L. Georgiadis, R. Guerin, and Y.Shavitt, "Improved fairness algorithms for rings with spatial reuse," *IEEE/ACM Transactions on Networking*, vol. 5, no. 2, pp. 190-204, April 1997.

[3] R. Simha, and Y. Ofek "A Starvation-free Access Protocol for Full-duplex Buffer Insertion Ring Local Area Networks," *Journal of Computer Networks and ISDN Systems,* vol. 21, 1991.

[4] V. Gambiroza, P. Yuan, L. Balzano, Y. Liu, S. Sheafor, and E. Knightly, "Design, analysis, and implementation of DVSR: A fair, high performance protocol for packet rings," *IEEE/ACM Transactions on Networking*, vol. 12, no. 1, pp. 85-102, February 2004.

[5] IEEE, IEEE Standard 802.17: Resilient Packet Ring (Draft Version 3.1), *http://ieee802.org/17*, February 2004.

[6] A. Shokrani, I. Lambadaris, J. Talim, "An Analytical Model for Fair Rate Calculation in Resilient Packet Ring," Carleton University Technocal Report, SCE-04-16, August 2004.